Modmobjam

_Jam tomorrow, jam yesterday, but also jam today_

By Sébastien Dudek

SSTIC RUMP 2018

June 14th 2018
Introduction

- Following Modmobmap presented at BeeRump 2018
- Helps to produce downgrade attacks as shown in House Intercoms Attacks presentations
- Uses Modmobmap results to jam mobile cells in a DIY way!
- Cheapest and efficient tricks to jam
Jam yesterday

With a portable/chinese device

- cheap
- jam the whole 2G/3G/(4G?) bands but requires some modifications
- poor signal

Desktop jammers
Jam yesterday

With a portable/chinese device

Desktop jammers

- heavy, cumbersome but powerful
- also needs a disabling to conserve rogue cells
Modifications on radio devices?! In 2018?
Jam today

- With Software-Defined Radio
- Many devices could be used even the cheapest:
  - bladeRF;
  - HackRF;
  - ADALM-PLUTO;
  - and so on.
Jam today

- With Software-Defined Radio
- Many devices could be used even the cheapest:
  - bladeRF;
  - HackRF;
  - ADALM-PLUTO;
  - and so on.

The bandwidth

KTHX! But how do you cover all frequencies with your toys bro?
## SDR specs

<table>
<thead>
<tr>
<th></th>
<th>HackRF</th>
<th>bladeRF</th>
<th>USRP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Radio Spectrum</strong></td>
<td>30 MHz – 6 GHz</td>
<td>300 MHz – 3.8 GHz</td>
<td>50 MHz – 6 GHz</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>20 MHz</td>
<td>28 MHz</td>
<td>16 MHz [2]</td>
</tr>
<tr>
<td><strong>Duplex</strong></td>
<td>Half</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td><strong>Sample Size (ADC/DAC)</strong></td>
<td>8 bit</td>
<td>12 bit</td>
<td>12 bit / 14 bit</td>
</tr>
<tr>
<td><strong>Sample Rate (ADC/DAC)</strong></td>
<td>20 MspS</td>
<td>40 MspS</td>
<td>64 MspS / 128 MspS</td>
</tr>
<tr>
<td><strong>Interface (Speed)</strong></td>
<td>USB 2 HS (480 megabit)</td>
<td>USB 3 (5 gigabit)</td>
<td>USB 2 HS (480 megabit)</td>
</tr>
<tr>
<td><strong>FPGA Logic Elements</strong></td>
<td>[4]</td>
<td>40k / 115k</td>
<td>25k / 75k / 150k</td>
</tr>
<tr>
<td><strong>Microcontroller</strong></td>
<td>LPC43XX</td>
<td>Cypress FX3</td>
<td>Cypress FX2</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>January 2014</td>
<td>Now</td>
<td>Now</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>$300 [6]</td>
<td>$420</td>
<td>$650</td>
</tr>
</tbody>
</table>

Solution: "Smart" jamming

In 3 steps:

1. scan cells with Modmobmap;
2. target an operator;
3. and jam only targeted channels;
Scanning with Modmobmap

Modmobmap recovers 2G/3G/4G and more cells pretty much like OsmocomBB monitor mode for 2G only.

```
$ sudo python modmobmap.py -m servicemode
=> Requesting a list of MCC/MNC. Please wait, it may take a while...
[+] New cell detected [CellID/PCI·DL_freq (83-6400)]
  Network type=4G
  PLMN=151515-1515
  Band=20
  Downlink EARFCN=6400
Found 5 operator(s)
{u'20810': u'F SFR', u'20820': u'F-Bouygues Telecom', u'20815': u'Free', u'20801': u'Orange F', u'20811': u'SFR Home 3G'}
[+] Unregistered from current PLMN
[+] New cell detected [CellID/PCI·DL_freq (f0e02-10787)]
  Network type=3G
  PLMN=208-1
  Band=1
  Downlink UARFCN=10787
  Uplink UARFCN=9837
=> Changing MCC/MNC for: 20810
[+] New cell detected [CellID/PCI·DL_freq (298-6400)]
  Network type=4G
  PLMN=208-10
  Band=20
  Downlink EARFCN=6400
[+] New cell detected [CellID/PCI·DL_freq (298-6300)]
  Network type=4G
  PLMN=208-10
  Band=20
  Downlink EARFCN=6300
[+] New cell detected [CellID/PCI·DL_freq (298-6200)]
  Network type=4G
  PLMN=208-10
  Band=20
  Downlink EARFCN=6200
```
Results

Unlike RE tools, it returns a JSON file with needed cells information to be reused with other tools ;)

```json
{
    "4b***−76": {
        "PLMN": "208−10",
        "arfcn": 76,
        "cid": "4b**",
        "type": "2G"
    },
    "60****−2950": {
        "PLMN": "208−20",
        "RX": 2950,
        "TX": 2725,
        "cid": 60***,
        "band": 8,
        "type": "3G"
    }
}
```

XGold BaseBands? → requires xgoldmon Modmodmap’s fork: https://github.com/FlUxIuS/xgoldmon
GnuRadio: playing with blocks

GnuRadio companion is really nice → can add, make, and remove blocks → generates Python code

Perfect to build the bases of our jammer. But we still need an idea of how to design the schema.
After many years of research...

Lot of experiments with #blockchains... and research and cool stuff WOW!
The formula

We have finally found THE formula!
And applied it on GnuRadio

Here is the final schema:

![](image)

The generated Python code was then edited to support the JSON input.
Results with a simple HackRF

Works pretty well when downgrading a call from 3G to 2G

But the number of cells to jam could raise the number of needed SDR devices.
Jam tomorrow

Could also be cheaper using *OsmoFL2k*

TODO

Some work is required target specific frequencies → right sample rate, carrier frequency and harmonics
Conclusion

Modmobjam:

- is a cheap way to jam mobile cells with only a phone and a HackRF
- but if cells to jam are important more SDR devices are needed
- the code will be published soon (throw away code recycled to something clean)

The Osmo-FL2K will be tested to use it as a jammer too.
THANK YOU FOR YOUR ATTENTION,

ANY QUESTIONS?